Investigate M2M-related communication standards that exist on the global market today

by

Aleksander Albretsen

Thesis in partial fulfilment of the degree of
Master in Technology in
Information and Communication Technology

Agder University College
Faculty of Engineering and Science

Grimstad
Norway

June 2006

Abstract

Most M2M applications use well-known communication technologies to interconnect the devices. Even though they use well-known communication technologies there are no widely used and well-defined M2M standards regarding the data exchange (application layer). This thesis investigates and identifies M2M related communication standards that exist on the global market today, and are applicable for M2M standardisation.

This thesis is limited to the following segments within M2M: Security, Automatic Meter Reading (AMR) and Utility Control. Today, and in the future, IP will play an important role within M2M. This thesis is therefore mainly focusing on standards that implement how to transfer the application layer using the IP-stack. M2M is defined in this thesis as an application with a central server communicating with end-devices through a gateway, using remote communication from server to gateway.

The following standards are investigated and found applicable in one or more of the selected segments: CIP, MODBUS, LonWorks, KNX, DLMS/COSEM, M-BUS, SIA, M2MXML, OPC and ZigBee. Each of the standards is explained within the thesis.

All standards are identified and categorised, and area of applications and proposed solutions are described. This thesis discusses the applicability regarding each segment, multiple services behind one gateway, bandwidth consumption, software update and interconnection of networks.

In conclusion, it is generally possible to create standardised M2M solutions based upon existing standards within the segments of Security, AMR and Utility Control. Some standards can be used as is, while others need to be used in combination with another standard to fit in to the M2M platform defined for this thesis. Utility Control and AMR has most suitable standards. The security segment needs more standardisation work to support full featured M2M based solutions.

Preface

This thesis is the partial fulfilment of my degree of Master in Technology in Information and Communication Technology at Agder University College (AUC), faculty of Engineering and Science. The work has been carried out in collaboration with Teleca Wireless Solutions in Grimstad, under the supervision of Arild Haglund from AUC and Ole Jonny Gangsøy from Teleca Wireless Solutions.

I would like to thank my supervisors Arild Haglund and Ole Jonny Gangsøy for all the help and support during this thesis. I would also like to thank Per Asbjørn Vestøl at AUC for the introduction to KNX and LonWorks, and Kim Tommy Humborstad at Smart Energy Applications in Grimstad for the introduction to ZigBee.

Grimstad, June 2006.

_____

Aleksander Albretsen

Index

Figures and Tables

1 - Introduction

This thesis investigates and identifies M2M related communication standards that exist on the global market today, and are applicable for M2M standardisation.

1.1 - Background

Most M2M applications use well-known communication technologies to interconnect the devices, such as GPRS and Ethernet, often using the IP protocol. Even though they use well-known communication technologies there are no widely used and well-defined M2M standards regarding the data exchange (application layer). There exists a wide range of standards for data exchange between computer devices today, e.g. HTTP used by a web-server and a web-browser to exchange HTML documents. This thesis investigates such communication standards, related to M2M, to identify if they are applicable for M2M standardisation.

The main purpose of this thesis is to give Teleca Wireless Solutions AS, herby Teleca, a better knowledge of which M2M related communication standards that exist on the global market today and are applicable for M2M communication standardisation.

Articles such as *The Impact of "Us Versus Them" Thinking* [32] by Harbor Research indicate the lack of standardisation within M2M communication. I believe that standards, and especially open standards, create a competitive and healthy market, because it limits the possibility for actors to create their own form of a monopolistic market. They are therefore significant since a competitive market can increase the quality of the products and services produced. In addition, Teleca believes that M2M communication needs standardisation.

1.2 - Delimitations

Since M2M is a wide definition, it is impossible to determine which standards that is applicable for M2M communication without any delimitation. It is therefore necessary to select segments within M2M to analyse. This thesis is limited to the following segments: Security, Automatic Meter Reading (AMR) and Utility Control. Each of these segments is described in chapter *2 - What is M2M?*.

Today, and in the future, IP will play an important role within M2M. This thesis is therefore mainly focusing on standards that implement how to transfer the application layer using the IP-stack.

1.3 - Thesis definition

The final thesis definition is as stated below:

> *Teleca believes that M2M communication needs standardization. The master thesis should investigate M2M-related communication standards that exist on the global market today, and identify protocols applicable for M2M communication standardization.*
>
> *The scope of the thesis should be narrowed to 3 segments within M2M:*
>
> *- Security segment*
>
> *- Automatic Meter Reading (AMR) segment*
>
> *- Utility Control segment*
>
> *The thesis should also discuss issues connected to communication bearers where charging is based upon amount of transferred data, i.e. GPRS and SMS. One or more communication standards may be implemented, if time allows it.*

1.4 - Report outline

This thesis will first introduce you to the concept called M2M by explaining each of the segments investigated, ending up with a definition for M2M in this thesis. Then all the applicable standards found are described one by one. Further, the applicable standards are organised using two different identification methods. After this, there is a chapter with area of application and proposed solutions, which starts the discussion. Then there are conclusions at the end.

2 - What is M2M?

M2M is in general an acronym for machine-to-machine, machine-to-man, man-to-machine, machine-to-mobile and mobile-to-machine. If you ask 10 persons what M2M is, you will probably get 10 different answers. This makes it difficult to have a common understanding of what M2M is. The article *Dispelling the Myth* [13] explains some of the aspects around M2M, but not all. Because of all this, it is necessary to define what M2M is within this thesis, giving us a common understanding. First, let us look at the segments within M2M that this thesis will deal with.

2.1 - Automatic Meter Reading (AMR)

Automatic Meter Reading (AMR) is a common name for all applications that automatically reads a meter and sends the result to a central server, providing data for billing and analysing. As an example for AMR solutions, we can use automatic reading of the power meter in a private household. This type of application enables the power provider to charge based upon the actual power consumed, and not an estimate based upon the consumptions over a longer period. The customer also gets more data about his or her power consumption, which enables better control.

Figure 2.1.1 shows the system components in an application for automatic reading of the power meter in a private household.

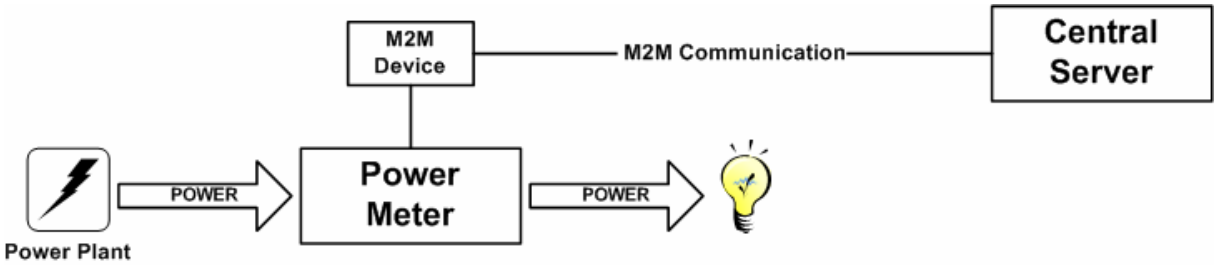

Figure 2.1.1 - AMR in a household, measuring the power consumption.

2.2 - Security

The security segment, within M2M, consists of all security applications that transfer data from a local monitoring system to a central server. A basic example for this is a home security system that reports its state to the service provider's central system, making it possible to report if a burglar is robbing your home.

One of the most central organisations within this segment is Securities Industry Association (SIA) [12].

Figure 2.2.1 shows a home security system with a M2M communication link between the *control unit* and the *central server*.

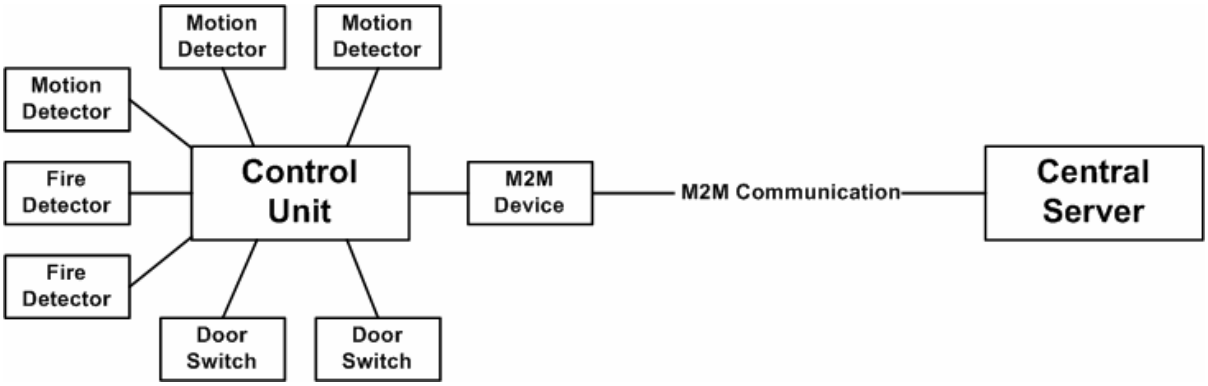

Figure 2.2.1 - Home security system with M2M communication

2.3 - Utility Control

Utility control, within M2M, enables remote surveillance and control of devices. As an example for this, we can use remote control of an advanced electrical system in a household.

Figure 2.3.1 shows such a system, existing of two lamps and a heater controlled by switches and a thermostat. *Switch1* controls *lamp1*, *switch2* controls *lamp2* and the *thermostat* controls the *heater*. All of the devices are monitored and manageable through the *control unit*. The control unit has a M2M device attached to it, which enables Utility Control.

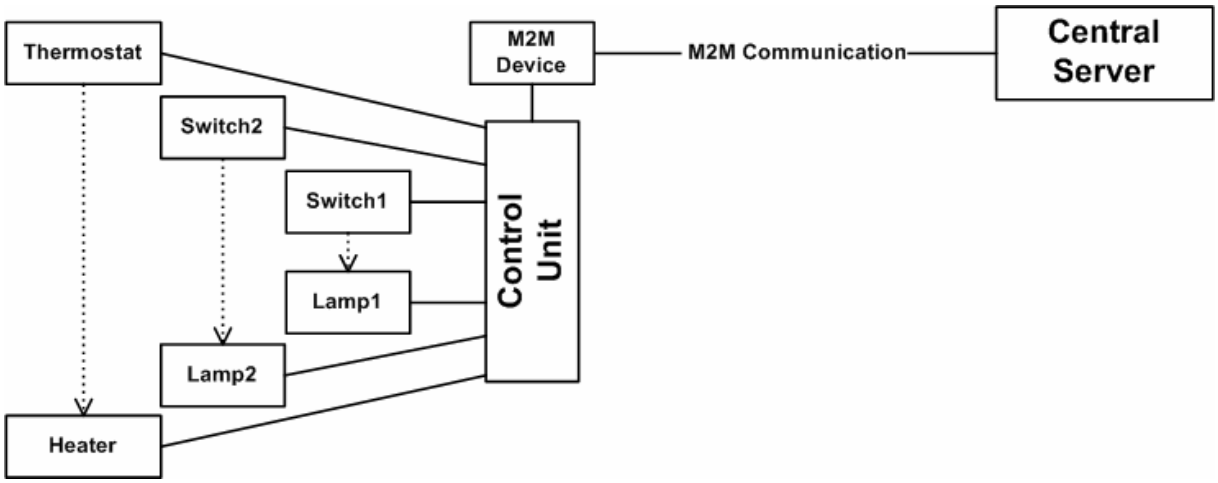

Figure 2.3.1 - Utility Control in a household

2.4 - M2M Communication

M2M communication is a communication link set up between to devices for remote communication. Technologies such as broadband connection, SMS, GSM, GPRS, EDGE and UMTS are therefore often suitable for this purpose.

Many of these technologies have the same downside of often charging based upon the amount of data transferred. This adds an extra cost factor to the M2M solution providers, since the bandwidth consumption will affect the total price of the solution.

2.5 - M2M in this thesis

The three covered segments can be generalised as a system with a M2M device making it possible to gather data and control devices inside the system, managed by a central server, using an M2M communication link. See figure 2.5.1 for more information.

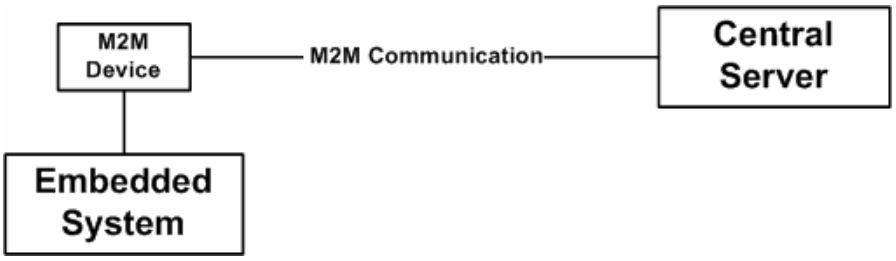

Figure 2.5.1 - Generalised M2M system

Combining this generalisation with the Teleca M2M platform [14] results in a architecture as shown in figure 2.5.2, and defines M2M in this thesis as a central server communicating, through a gateway, with end-devices.

Derived from the Teleca M2M platform the server-to-gateway communication uses communication bearers such as broadband connection, SMS, GSM, GPRS, EDGE or UMTS. The gateway-to-device communication is a local end-device network.

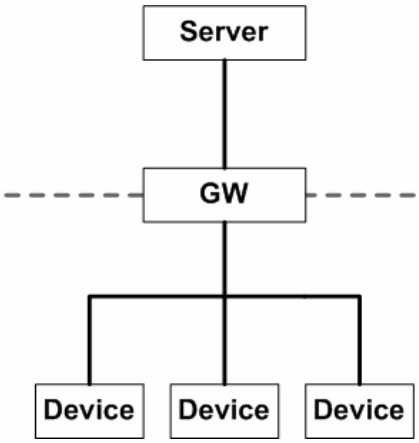

Figure 2.5.2 - General M2M architecture in this thesis

3 - Applicable standards

The applicable standards are found looking into solution providers within the selected segments and industrial automation standards. This chapter describes them one by one.

3.1 - CIP (DeviceNet, ControlNet and Ethernet/IP)

CIP [7], Common Industrial Protocol, is an industrial standard for automation systems maintained by ODVA [15] (Open DeviceNet Vendor Association) and ControlNet International [16]. As the name reveals CIP defines a common application and transport layer, which is currently used in DeviceNet, ControlNet and Ethernet/IP [8]. DeviceNet and ControlNet are two different industrial networks and Ethernet/IP is a specification that describes how to encapsulate and transport CIP using IP. DeviceNet, ControlNet and Ethernet/IP devices can therefore interconnect using network gateways.

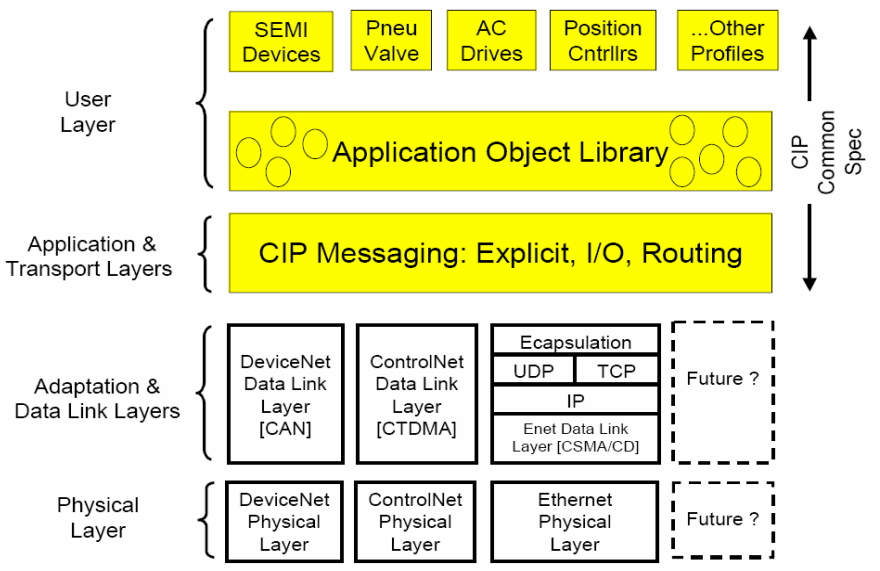

Figure 3.1.1 - CIP Architecture and Related Specifications, from [8]

CIP is an object-oriented protocol, which primarily connects sensors and actuators to their controllers. In addition to transport control data from sensors and to actuators, CIP can also transport other information such as configurations and general information. CIP supports point-to-point, multicast and peer-to-peer communication.

CIP is structured using the following physical organisations:

System = { Domian(s) } Domain = { Network(s) } Network = { Subnet(s) }
Subnet = { Node(s) } Segment = { Node(s) } Node = { Object(s) }

Each CIP node is modelled as a set of objects describing the connected sensors and/or actuators. Each object model describes attributes, services and the behaviour of an object. The CIP specification contains an object library including almost all types of sensors and actuators. Each implementation of CIP, e.g. Ethernet/IP, has its own set of implementation specific objects. In addition, it is also possible do describe and use proprietary object models.

3.2 - MODBUS

Modbus [26] is an open application protocol for master-slave/client-server communication developed by Modicon. It is a request/response protocol used for changing or retrieving properties from the server to the client. It is currently implemented using TCP/IP over Ethernet, asynchronous serial transmissions (RS232, 422, 485, etc.) and MODBUS PLUS (high-speed token passing network). The protocol is relatively simple and easy to implement and is there for widely used within many different industrial environments.

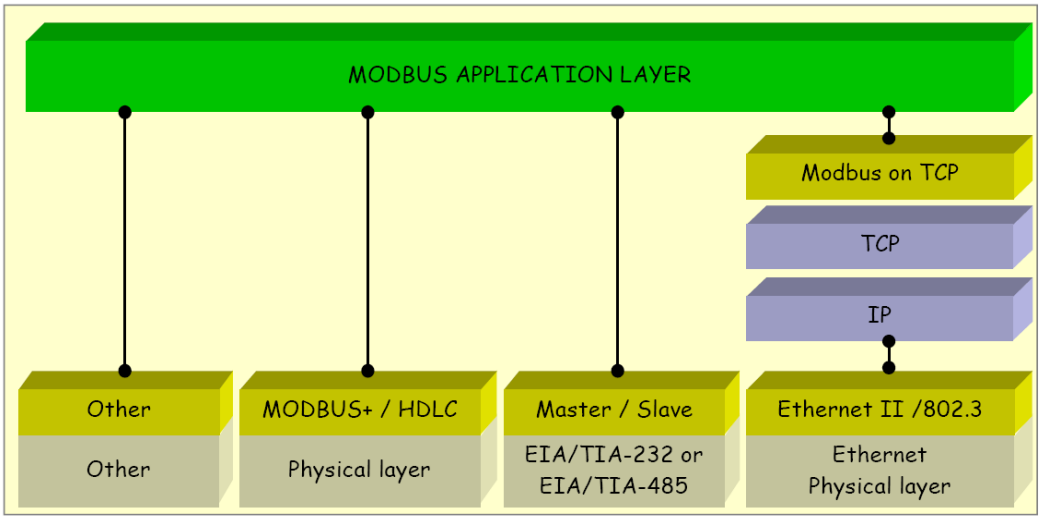

Figure 3.2.1 - MODBUS Application Layer, from [26]

Modbus defines a simple protocol data unit (PDU), which is independent of the underlying protocol layers. A modbus PDU consist of a function code and a data field. Each modbus implementation defines an application data unit (ADU) which adds address field and error check, see figure 3.2.2. The size of a modbus PDU is limited to 253 bytes because of the first modbus implementation using a serial line (RS485) which has a maximum ADU length of 256 bytes. One byte is used for addressing and two for error check (CRC), which leaves us with 253 bytes left for the PDU.

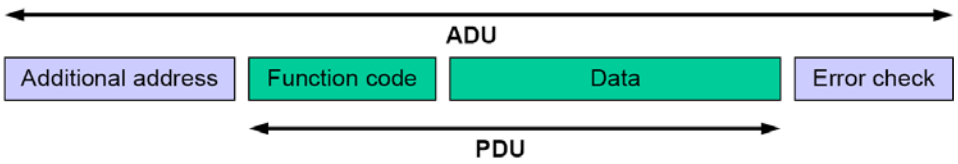

Figure 3.2.2 - General MODBUS frame, from [26]

The modbus data provided by a server is sorted in four primary categories, each containing 65536 different values that can be individually addressed (0000_{hex} - $FFFF_{hex}$):

- Discretes Input: single bit, read-only (collection of bits only readable by the client)
- Coils: single bit, read-write (collection of bits which the client can alter)
- Input Registers: 16-bit word, read-only (collection of *words* only readable by the client)
- Output Registers: 16-bit word, read-write (collection of *words* which the client can alter)

Modbus defines a set of functions used by the client to make a request to the server, see figure 3.2.3 for a complete function list. For each function there is defined a request format, a response format and an error format. As an example, we can use the specification for the function *0x01 - Read Coils* shown in figure 3.2.4. Figure 3.2.5 shows an example of modbus data traffic where the client asks the server for the data stored in coil 20 to 38.

				Function Codes		
				code	*Sub code*	*(hex)*
Data Access	**Bit access**	Physical Discrete Inputs	Read Discrete Inputs	**02**		**02**
		Internal Bits Or Physical coils	Read Coils	**01**		**01**
			Write Single Coil	**05**		**05**
			Write Multiple Coils	**15**		**0F**
	16 bits access	Physical Input Registers	Read Input Register	**04**		**04**
		Internal Registers Or Physical Output Registers	Read Holding Registers	**03**		**03**
			Write Single Register	**06**		**06**
			Write Multiple Registers	**16**		**10**
			Read/Write Multiple Registers	**23**		**17**
			Mask Write Register	**22**		**16**
			Read FIFO queue	**24**		**18**
	File record access		Read File record	**20**	**6**	**14**
			Write File record	**21**	**6**	**15**
Diagnostics			Read Exception status	**07**		**07**
			Diagnostic	**08**	**00-18,20**	**08**
			Get Com event counter	**11**		**OB**
			Get Com Event Log	**12**		**0C**
			Report Slave ID	**17**		**11**
			Read device Identification	**43**	**14**	**2B**
Other			Encapsulated Interface Transport	**43**	**13,14**	**2B**

Figure 3.2.3 - MODBUS function list, from [26]

Request

Function code	1 Byte	**0x01**
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of coils	2 Bytes	1 to 2000 (0x7D0)

Response

Function code	1 Byte	**0x01**
Byte count	1 Byte	**N***
Coil Status	**n** Byte	n = N or N+1

*N = Quantity of Outputs / 8, if the remainder is different of 0 ⇒ N = N+1

Error

Function code	1 Byte	**Function code + 0x80**
Exception code	1 Byte	01 or 02 or 03 or 04

Figure 3.2.4 - MODBUS function definition example, from [26]

Request		**Response**	
Field Name	*(Hex)*	*Field Name*	*(Hex)*
Function	**01**	Function	**01**
Starting Address Hi	**00**	Byte Count	**03**
Starting Address Lo	**13**	Outputs status 27-20	**CD**
Quantity of Outputs Hi	**00**	Outputs status 35-28	**6B**
Quantity of Outputs Lo	**13**	Outputs status 38-36	**05**

Figure 3.2.5 - MODBUS data trafic example, from [26]

3.3 - LonWorks, NES and Pyxos

LonWorks, often address as lon, is a complete architecture for automation systems developed and owned by Echelon [1]. Lon is today mostly used for AMR and building automation.

The network architecture in lon is based on the Local Area Network (LAN) architecture. But they have some important differences. A lon network is a low speed control network with the ability to carry many transactions per second. In a lon network at 1.25Mbps there can be up to 700 transactions per second. This differs from the best effort design used in LANs. A complete lon network consists of routers and gateways, which physically divide the network in to segments. A lon router is “intelligent” and only forwards a package to the needed segment. In addition to this, a lon network is also logically separated using groups, subnets and domains. Lon supports up to 2^{48} independent domains, a domain is a collection of up to 255 subnets with up to 127 nodes in one subnet. This makes lon capable of addressing 32.385 nodes per domain. Subnets cannot cross routers. In addition to this, nodes can be grouped together across subnets within a domain with a maximum of 256 groups in one domain. One lon node can be a member of up to 15 different groups at the same time.

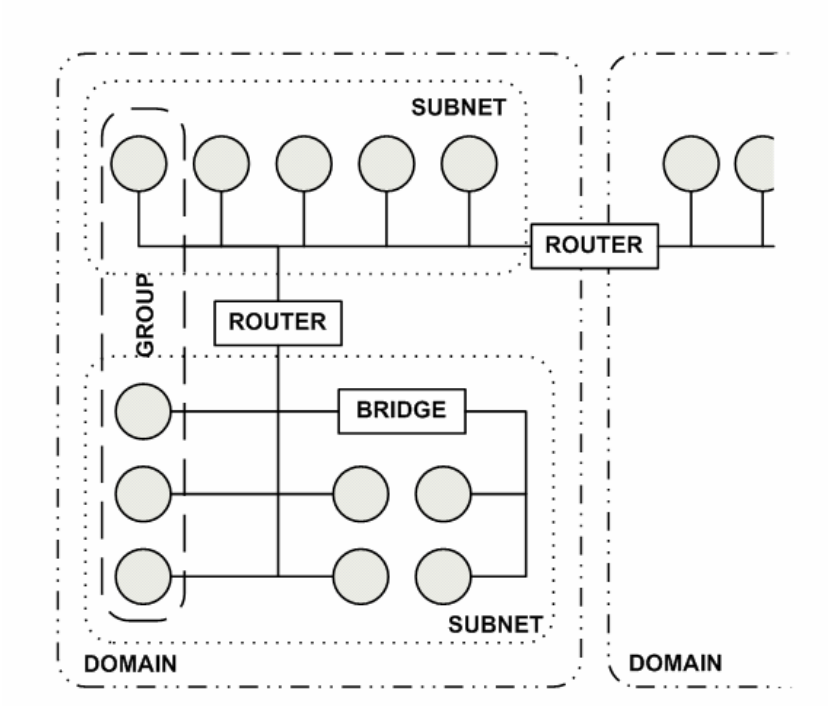

Figure 3.3.1 - Example of LonMark object model

The lon communication standard is named LonTalk® [2, 3] and covers all the seven OSI layers. In brief, a lon network consists of sensor objects, controller objects and actuator objects. One node can consist of one or more of these objects. Each object has a set of input and/or output signals called SNVTs (Standard Network Variable Types). Each type of output signal can be connected to the same type of input signal at another object, see figure 3.3.2. In addition, each node can be configured using SCPTs (Standard Configuration Parameter Types).

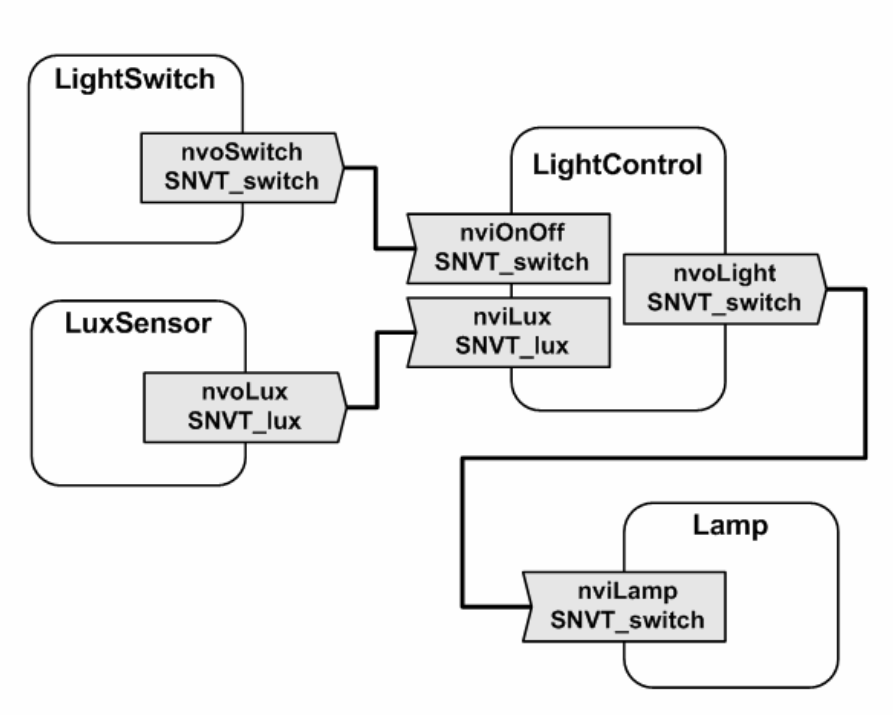

Figure 3.3.2 - LonTalk® model example

Each node in a lon network contains a Neuron® [2, 3] chip. A neuron chip consists of three processors that deal with different layers of the OSI model. One processor for media access (layers 1 and 2), one for the network layers (3 to 6) and one for the application (layer 7). The software running in the two first processors is “pre” written, so the lon device developer only needs to implement an application. There is a wide range of neuron chips available making lon support twisted par, coaxial, fiber optics, powerline and RF based networks.

Lon also specifies how to transfer the LonTalk protocol using IP. The Echelons i.Lon product series [6] provides this gateway functionality, and enables the use of lon networks in M2M applications. The i.Lon 100 Internet Server has an option for external GPRS modem.

NES

NES (Networked Energy Services) [4] is Echelon’s AMR solution. This solution consists of meters connected to a data concentrator through a powerline network. A computer with NES system software connects to the data concentrator using IP to retrieve the meter data. The entire solution is based upon the LonWorks technology.

Pyxos

Pyxos [5] is a wired sensor network solution, from Echelon, designed to compete with ZigBee [27]. As an example, we can use a vending machine. Each sensor, controller and actuator within the vending machine is then a pyxos node, connected together in a pyxos network. This pyxos network can also contain a lon gateway providing the ability to connect all the vending machines together in a lon network.

3.4 - KNX (EiB)

The KNX [22] standard is a specialised form of automation system designed for building applications. It is a result of a formal merge between BatiBUS, EIB, and EHS. KNX is based on EIB, which makes EIB upwards compatible with KNX. The Konnex Association [23], which also provides certification, training and software tools, maintains the KNX standard.

KNX supports three configuration mechanisms: System (S)-mode, Easy (E)-mode and Automatic (A)-mode. A-mode is intended for appliances where uninstructed users can set the system into operation without the need of any configuration tool, in a plug-and-play like fashion. In a-mode, all appliances know how to set up the bindings to each other, and addresses are dynamically assigned. E-mode is intended for easy installation without the need of advanced tools, but the devices does not set up the bindings them selves, it has to be configured by the user. S-mode is the complete system mode containing configuration tools used by engineers to plan, engineer and configure a KNX system.

KNX is designed as a network with the ability to route the communication traffic. KNX supports 65.536 units using a 16-bit individual address field. The logical network topology supports 255 devices on one line. Up to 16 lines can be connected together with a main line into an area, and all the areas are grouped together by a backbone line forming a complete KNX domain of up to 16 areas, as shown in figure 3.4.1.

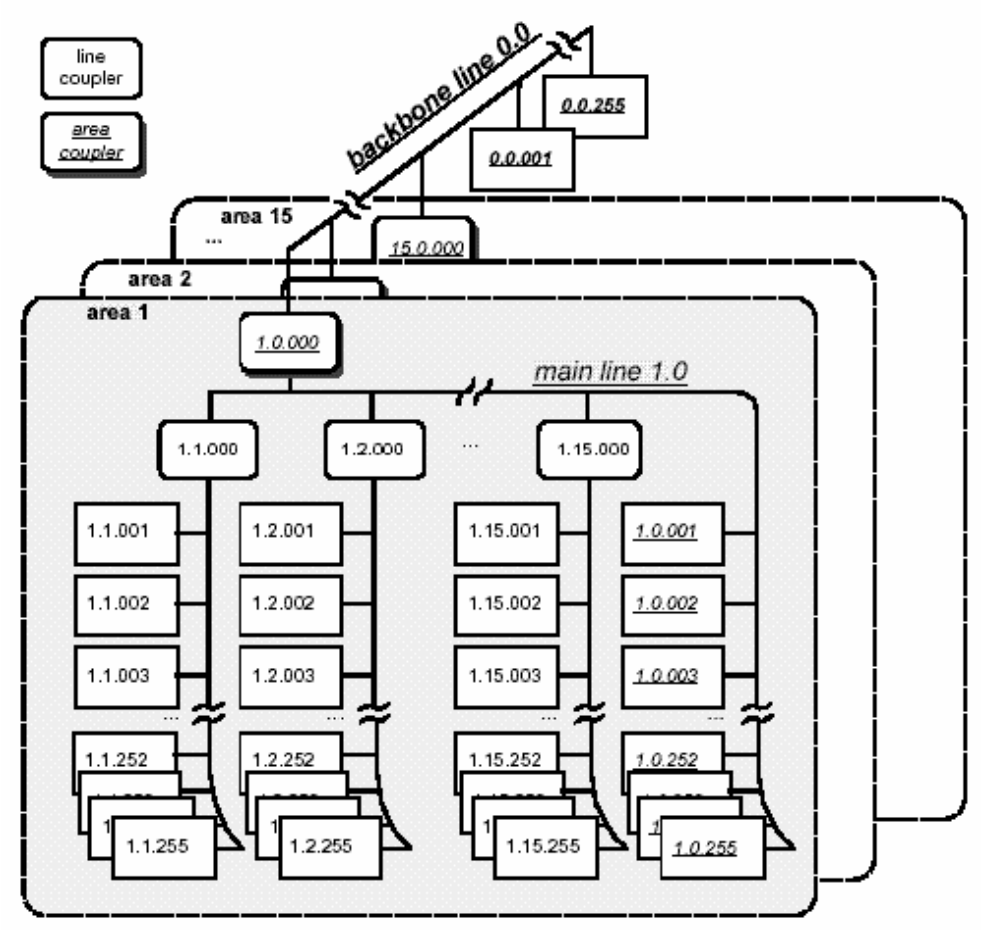

Figure 3.4.1 - The logical topology of KNX, from [22]

KNX supports multicast, broadcast and both connection-oriented and connection-less one-to-one communication.

KNX devices can communicate with each other using the following communication mediums:

- TP0, twisted pair network inherited from BatiBUS
- TP1, twisted pair network based on the basic medium of EIB
- PL110, power line communication (plc) inherited from EIB
- PL132, power line communication (plc) inherited from EHS
- RF, radio network fully specified within KNX
- IR, infrared communication inherited from EIB
- IP, fully specified within KNX how to enable IP communication.

3.5 - DLMS / COSEM

Together DLMS [18] and COSEM [18], also known as IEC 62056, is a modern object-oriented standard for automatic meter reading (AMR).

DLMS	Device Language Message Specification	Application layer specification
COSEM	COmpanion Specification for Energy Metering	Object-oriented interface model.

DLMS User Association [20] (DLMS-UA) is the organisation maintaining the standards, and they provide conformance testing as described in their yellow book [19].

COSEM is designed to provide a more up to date object oriented approach, but still backward compatible with the existing DLMS standard. COSEM therefore includes a new version of DLMS called xDLMS. xDLMS is the application layer providing access to the COSEM objects.

In DLMS/COSEM the metering equipment acts as a server, end the data collection system as a client. This may seem a little odd at first, but can perfectly be explained by the fact that a meter provides a service, meter data, that a collection system wants. In other words the collection system connect to the meter and makes use of its services. A meter can handle connections from multiple collection clients, and a collection client can connect to multiple meters. However, a meter cannot connect to another meter. Each meter provides a set of COSEM objects that a collection system can access through xDLMS.

Figure 3.5.1 shows an example of a DLMS/COSEM system where both the meters (server) have connections from multiple collection systems (client), and collection systems connecting to multiple meters.

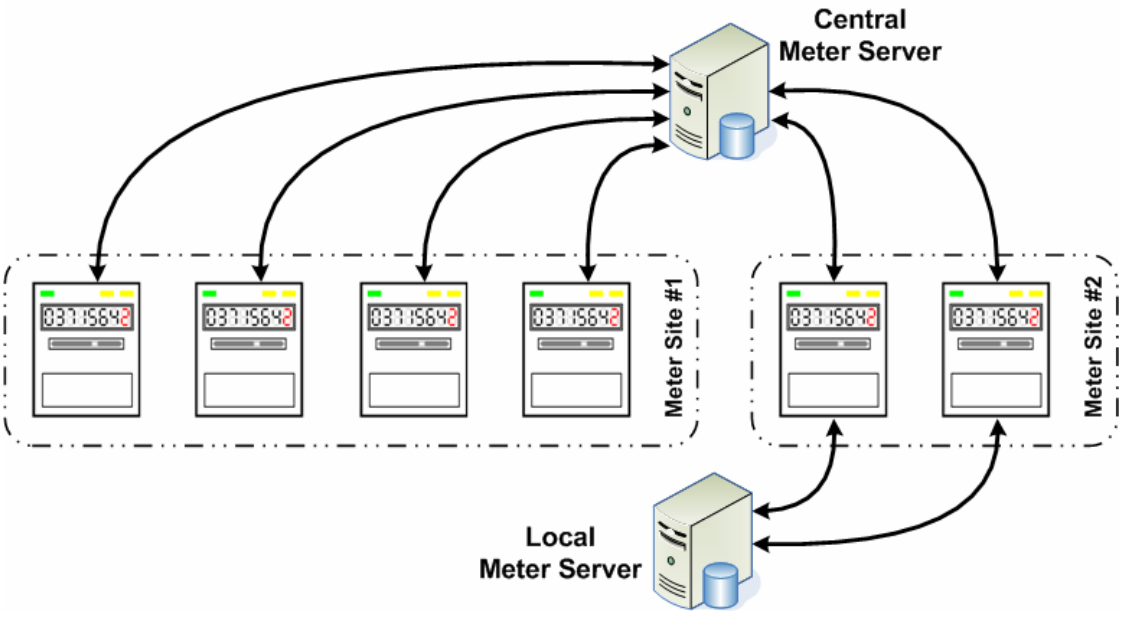

Figure 3.5.1 - Example of logical DLMS/COSEM architecture.

Since DLMS is an application layer protocol, it can be transferred using different network technologies. DLMS-UA has currently specified how to use HDLC [21], TCP/IP and UDP/IP. Figure 3.5.2 shows a DLMS/COSEM system using both HDLC and IP with gateways.

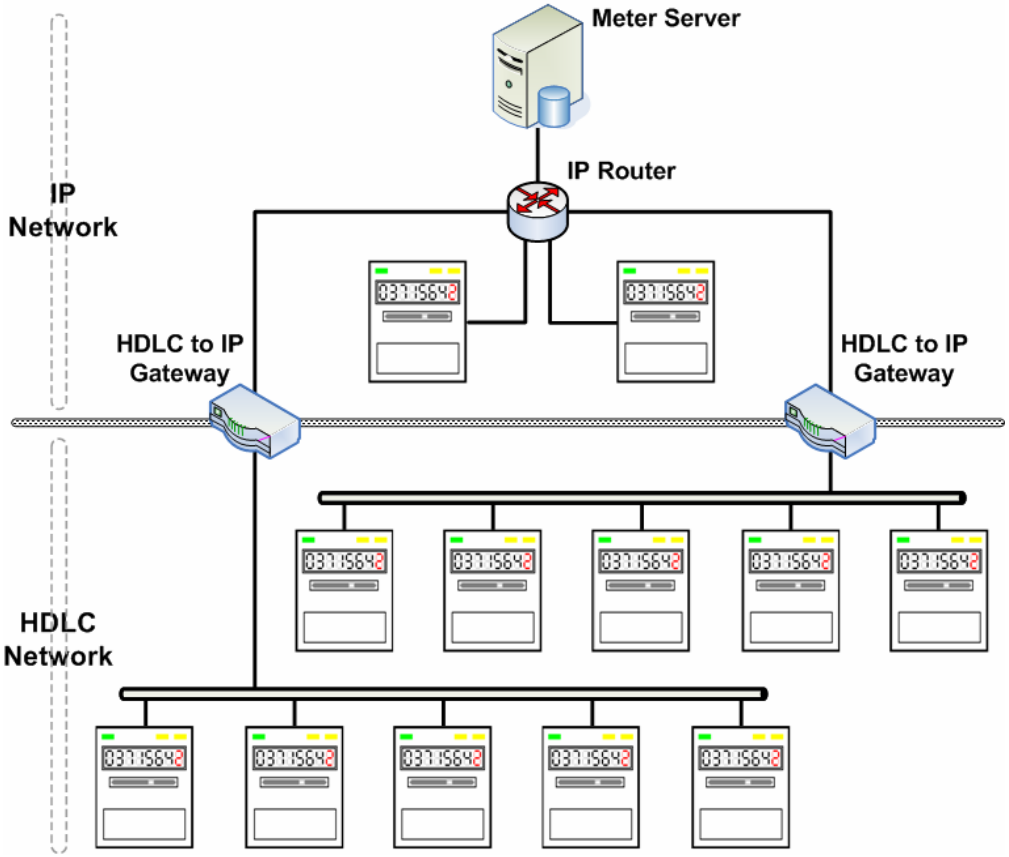

Figure 3.5.2 - DLMS/COSEM example with both IP and HDLC

3.6 - M-Bus (Meter Bus)

M-bus [17] is a low cost home electronic system (HES) designed to fill the need for networking and remote reading of utility meters such as a gas, water and power meters.

M-bus is a bus system, not a network, and therefore are many of the osi-layers left unused. As shown in table 3.6.1 m-bus specifies its own physical layer and relate on other standards for the data link and application layer.

Layer	**Function**	**Standard**
Application	Data structures, data types, actions	EN1434-3
Presentation	empty	
Session	empty	
Transport	empty	
Network	extended addressing (optional)	
Data Link	transmission parameters, telegram, formats, addressing, data integrity	IEC 870-5
Physical	cable, bit representation, bus extensions, topology, electrical specification	M-Bus

Table 3.6.1 - M-bus and the OSI model, from [17]

An m-bus network consists of one master with a maximum of 250 slaves (meters). M-bus supports remote powering of the meters through the bus. M-bus also support multiple addresses per slave and multicast.

The data link layer in m-bus is derived from IEC 870-5, but does not use all of its features. The slaves in an m-bus system cannot communicate with each other.

The application layer in m-bus is derived from the application layer in EN1434-3, which is designed for use with heat meters, but it is also suitable for other meters like power and water meters.

Since m-bus supports baud rates from 300 to 38400 baud, it has an additional layer to the seven OSI layers. This layer is needed since the osi-model does not allow changing of baud rate and address from higher layers. This management layer allows an m-bus node to change its baud rate.

3.7 - SIA and similar standards

Within the security segment DTMF, or DTMF like methods, is widely used to report an alarm or event from a building or unit to a central system. There is a wide range of proprietary standards used, since almost all dominating manufactures has their own standard. SIA [25] (Securities Industry Association) defines the most well known open standards.

Whether it is a SIA protocol or a proprietary protocol, they all use the same methodology. The remote device is attached to an analogue telephone line and has a predefined phone-number that it calls up when it has an alarm or event to report. The central system responds to the call and synchronise with the calling unit. Upon established communication, a predefined bit sequence is transferred. The synchronisation, physical method used (often DTMF) and how the bits are arranged varies according to which standard used.

In “modern” applications, GSM or ISDN may be used to establish the analogue communication channel. Some also transfer the same bit sequence coded within a SMS or an IP-package. There is no standard that defines this, only proprietary solutions are used to gain “SIA over SMS” or “SIA over IP”.

3.8 - M2MXML

M2MXML [9] is a lightweight open standard for M2M communication based upon XML, initially developed by Sensor Logic [10]. M2MXML is now a Source Forge [11] project, meaning that whom ever that is interested can participate in the project.

M2MXML is a communication standard that describes how to format messages exchanged between two M2M devices using XML. It consists of a set of pre-defined commands and their attributes. M2MXML does currently not describe how to connect to devices and how to exchange the message. Neither does it provide a conformance programme to verify compatibility. This makes it difficult to get M2M devices supporting M2MXML to interconnect with each other.

Figure 3.8.1 shows and example of a command sent with M2MXML. The command instructs a controller for an electrical system to turn on the device ‘lamp243’.

```
<M2MXML ver="1.0">
  <Command name="turnOn" address="lamp243" seq="321" />
</M2MXML>
```

Figure 3.8.1 - M2MXML sample message

3.9 - OPC

OPC [29], OLE for Process Control, maintained by the OPC Foundation [30] consist of a collection of standards for open connectivity. OPC got its name because of the first and originally OPC standard based upon Microsoft OLE, today known as the Data Access specification. The OPC Foundation also provides certification service and interoperability test tools.

OPC is a complete architecture for field, process and business management providing a common communication platform concentrating on data access and not the types of data, see figure 3.9.1. However, it is possible and quite common to use only parts of OPC. Each part of the OPC specification is described in separate sections below.

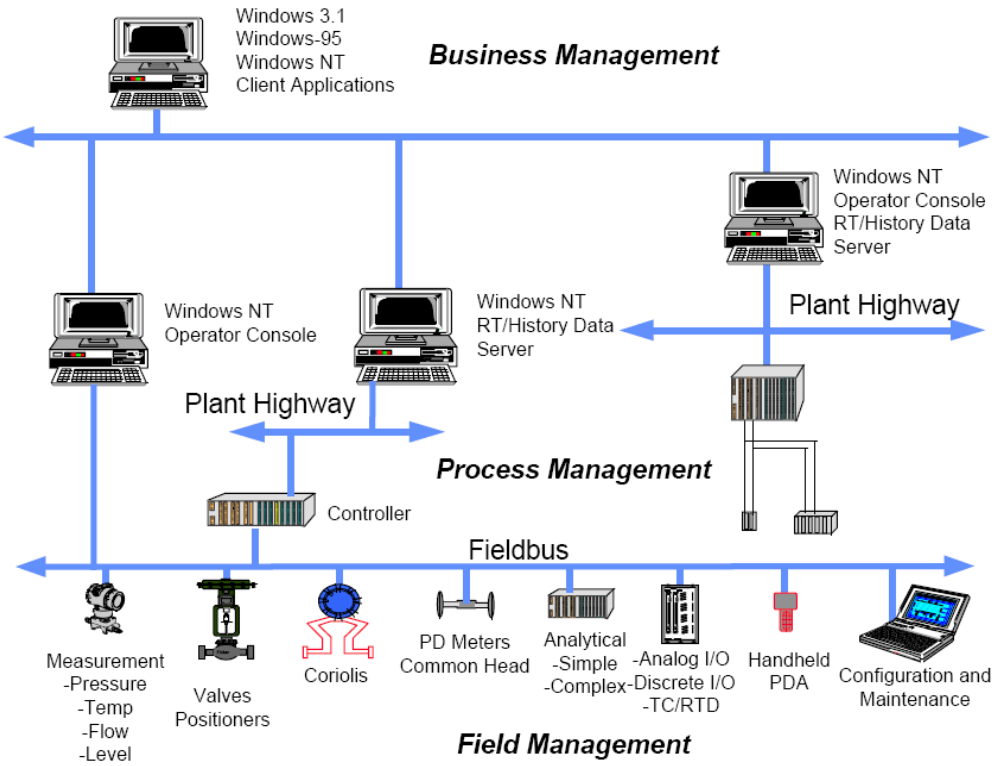

Figure 3.9.1 - OPC: Process Control Information Architecture, from [31]

OPC Data Access

OPC data access servers consist of three types of objects: the server object, group objects and item objects. The server object contains information about the server and serves as a container for group objects. Each group object contains information about the group and serves as a container for item objects, making it possible to form a sorted data structure. Each item consists of a value, a quality label and a time stamp. The client cannot access an item directly; it has to get the information from a group object. It is therefore not defined any external interfaces for OPC items.

OPC Data eXchange

OPC data exchange specification extends the OPC data access specification providing server-to-server and client-to-client data access. In other words, it defines how to exchange data between two OPC data access servers. It also provides an interface for remote configuration of the exchange server.

OPC Alarm & Event

OPC states an alarm as an abnormal condition, thus a special case of condition. When reached, it produces an event. OPC also support events not associated with a condition such as operator actions and system errors. This specification provides interfaces for a client to browse the available events and subscribe to them. When an event occurs the OPC server informs the client.

OPC Historical Data Access

Over time, a process produces historical data. This specification defines interfaces making a OPC client able to access this information from the OPC server. OPC supports everything from simple trend servers to complex analysis servers that may provide average, minimum and maximum values, etc.

OPC Batch

The OPC batch specification provides common interfaces for exchanging data in batch processing industry based upon ANSI/ISA-S88.

OPC XML-DA

OPC xml-da defines how plant floor data can be described using XML.

OPC Complex Data

OPC complex data provides data access and xml-da with the ability to describe complex data such as binary structures and XML structures.

OPC Security

OPC security specification states how to secure the data in an OPC server to prevent unauthorised access to the process data. The security working-group is also responsible for addressing issues regarding security in the existing OPC specifications.

OPC Commands

OPC has formed a new working group that will provide interfaces for OPC servers and clients to identify, send and monitor commands that execute on a device.

3.10 - ZigBee

ZigBee [27] is a new up and coming standard for wireless sensor networks maintained by the ZigBee Alliance [28]. ZigBee is designed to support a large number of interconnected low power battery driven devices. As figure 3.10.1 shows it is build upon IEEE 802.15.4 and supports user defined application layers as well as ZigBee Alliance defined profiles.

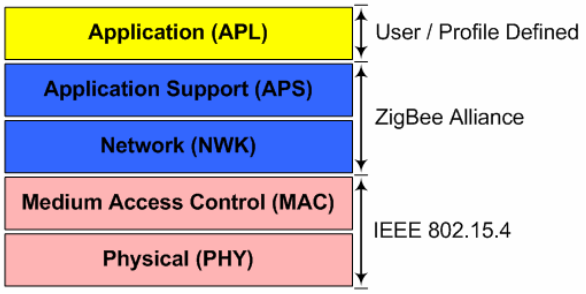

Figure 3.10.1 - ZigBee Stack

ZigBee defines two types of devices: Full Function Device (FFD) and Reduced Function Device (RFD). A full function device is often a device with good access to power and can therefore act as a coordinator, a router or an end-device if desirable. A reduced function device is often battery powered and can therefore only act as an end-device.

The network layer (NWK) defines three types of topologies: star, tree and mesh. Star networks consist of one ZigBee coordinator and end-devices. The coordinator is responsible for initiating and maintaining the network and all the end-devices. Each end-device only communicates with the coordinator, which makes the coordinator active in all communication as it forwards traffic from one end-device to another. In tree and mesh networks the coordinator is still responsible for initiating and maintaining the network, but FFD can become routers and extend the network. A new device will therefore be able to connect to the network without direct communication access to the coordinator as long as it reaches a router. Each router (or the coordinator) with its end devices is a local star implementation. In a tree network, these stars are organised in a tree structure with the coordinator as the root. In a mesh network the connection between the stars are organised in a peer-to-peer manner.

An end-device will always search for a coordinator or a router when it connects to the network. The end-device does therefore not need to know if it is a star, tree or mesh network it connects to, since it only acts in accordance with its master (coordinator or router).

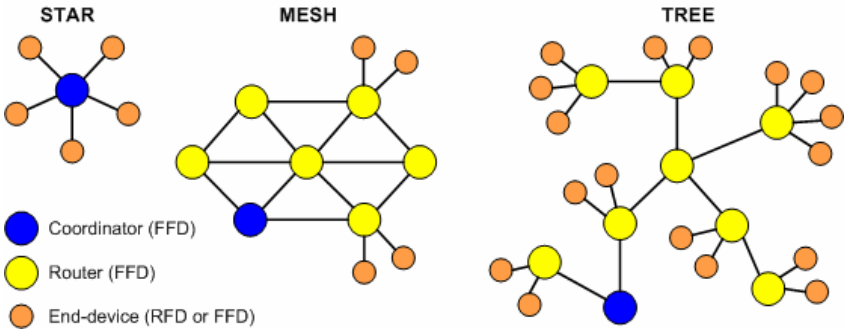

Figure 3.10.2 - ZigBee Network Topolgies

ZigBee application objects are logically connected using bindings, and support two main methods. The binding table can either associate two devices directly or through the coordinator. If a binding is attached directly between two devices, the coordinator has no knowledge of when the state of an object changes and can therefore not notice anyone of the change.

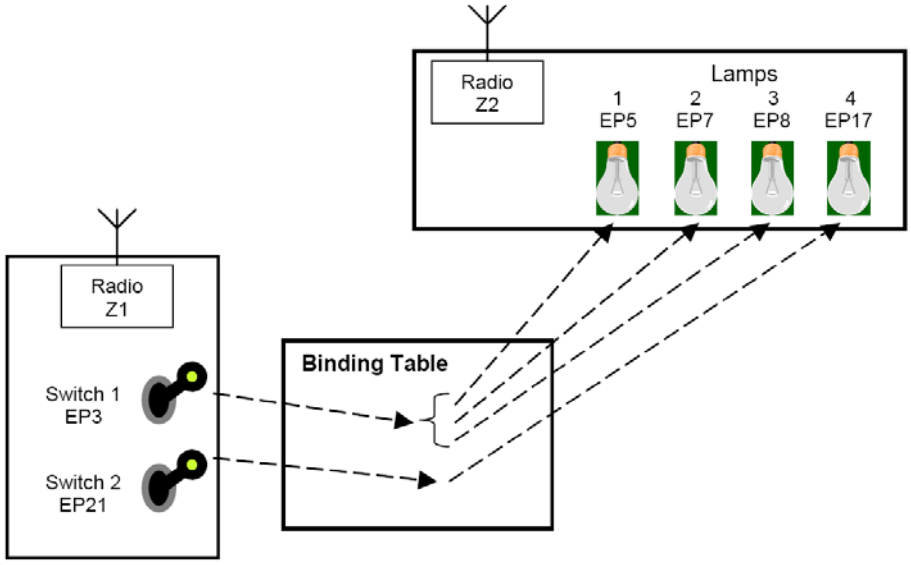

Figure 3.10.3 - ZigBee Binding and Binding Table, from [27]

Each ZigBee node has its unique MAC address. In addition to this address, each device is assigned a network specific address, called short-address, which is 16 bit long. This makes ZigBee applicable to create a network of over 65.000 nodes.

For the time being, there are defined four stack profiles within the ZigBee specification. One general profile used for proprietary solutions and profiles for Home Control, Building Automation and Plant Control. Within each of these stack profiles there can be defined application profiles. There are currently only one such profile defined, the Home Controls-Lightening profile defined using the Home Control stack profile. Such profiles make ZigBee a complete standard covering all the seven osi-layers, enabling interoperability between appliances from different vendors.

4 - Identification

4.1 - Architectural

The applicable standards can be divided in to the following three groups to ease the understanding of the architectural structure:

(a) Standards which provides a specification of how to transport the application layer using different underlying protocols, providing the ability to access a device directly through a transparent gateway.

(b) Standards design for information exchange using IP.

(c) Standards designed for local device-networks, such as sensor networks.

By combining (b) and (c) it is possible to make an architecture that looks similar to (a). But, the application layer is terminated and translated in the gateway.

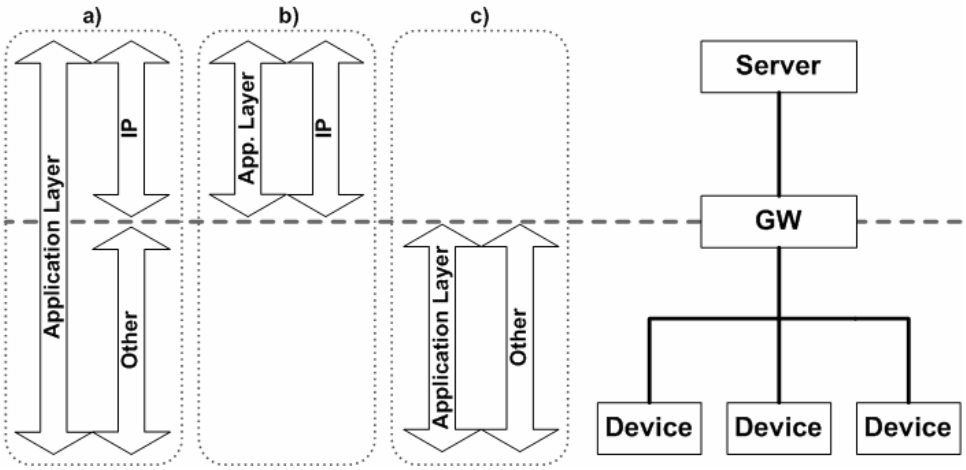

Figure 4.1.1 - Architecture Model

Table 4.1.2 below states where the standards belong. MODBUS is included both in (a) and (c) because it really is a category (a) standards. However, it is often used as a category (c) standard, which also is where it is most suitable for M2M applications.

Standard	a)	b)	c)
CIP	x		
MODBUS	x		x
LonWorks	x		
KNX	x		
DLMS / COSEM	x		
M-BUS			x
SIA			x
M2MXML		x	
OPC		x	
ZigBee			x

Table 4.1.2 - Architectural Grouping

4.2 - Horizontal

Each of the applicable standards is constructed to cover one or more vertical application. The model below, table 4.2.1, describes which vertical application each of the standard supports. This makes a horizontal picture of which standard to use in different applications.

The light grey box describes the applications supported when the standard is used alone.

The dark grey box describes the applications supported by combining a server-to-gateway standard and a gateway-to-device standard. For example by combining M2MXML and M-BUS, using M2MXML from server to gateway and M-BUS from gateway to device, providing architecture for AMR applications.

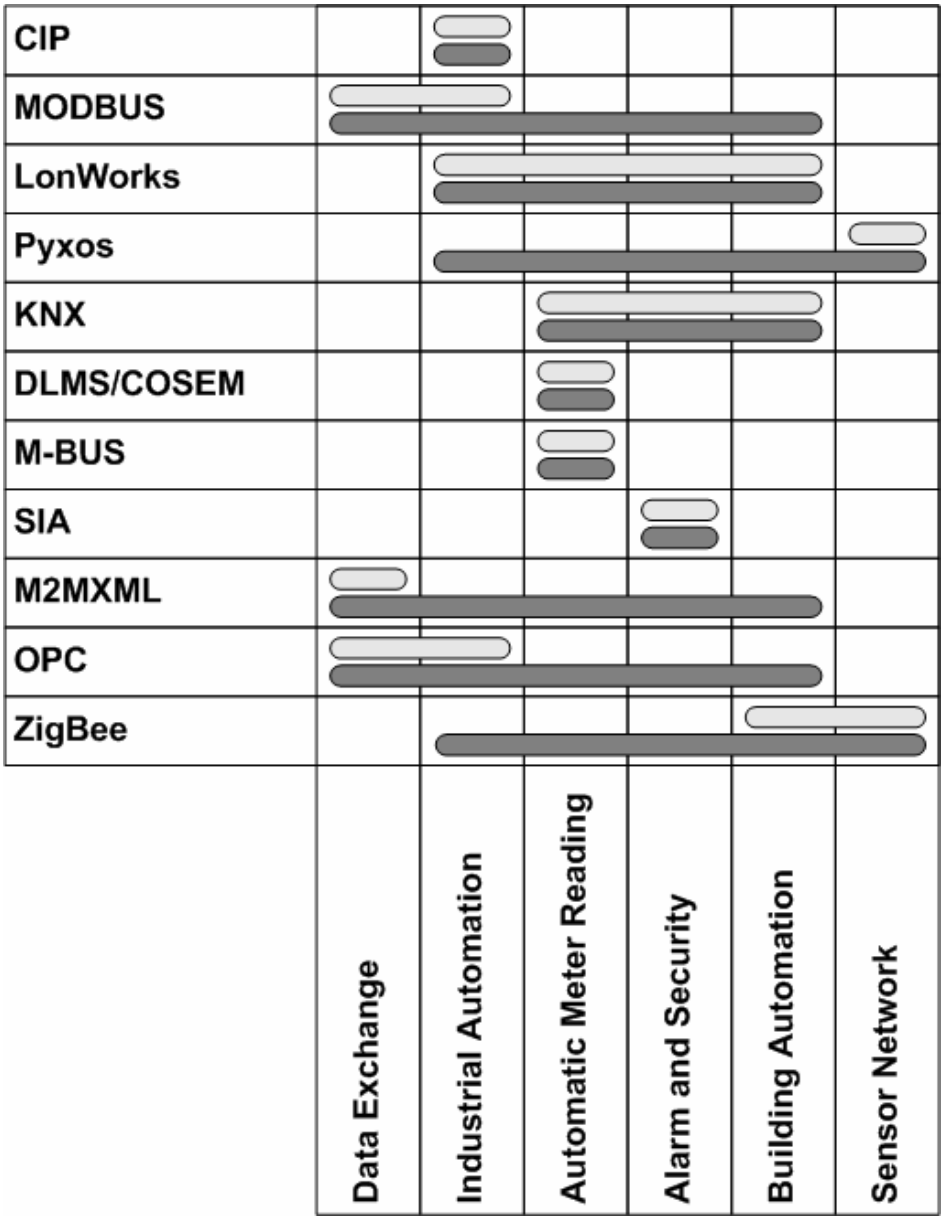

Table 4.2.1 - Application Coverage

5 - Area of application and proposed solutions

Derived from the definition of M2M in this thesis there are two main are of application that these standard will be used in regarding M2M. See figure 5.1.

The first area of application, called *full knowledge*, is a server-gateway-device application where the M2M solution provider has full knowledge about the system, and the ability to select which standards to use in both the server-to-gateway and gateway-to-device communication. There are two types of *full knowledge* applications, which differs whether the gateway is transparent or not.

The second area of application, called *black box*, is a server-gateway application where the M2M solution provider has semi knowledge about the system, and only the ability to select which standards to use in the server-to-gateway communication.

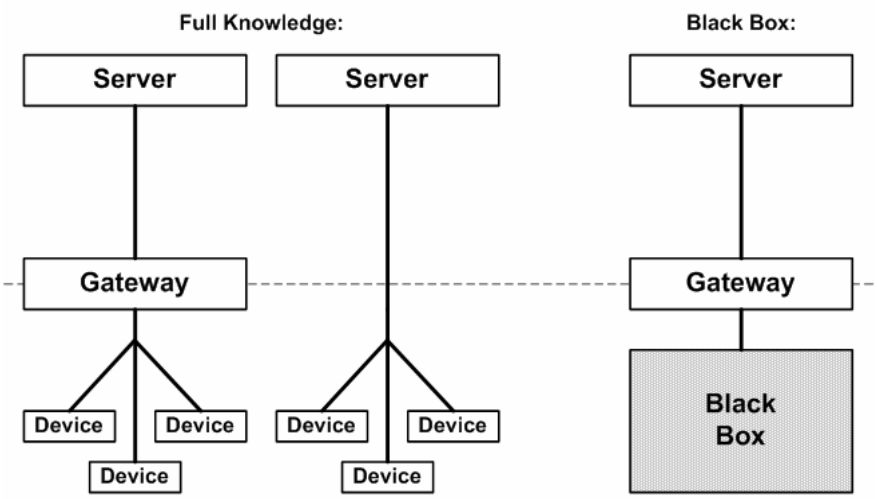

Figure 5.1 - Main types of solutions

Based on the information in chapter *4 - Identification*, the standards CIP, LonWorks, KNX and DLMS/COSEM are generally suitable for *full knowledge* applications with transparent gateway, and OPC, MODBUS and M2MXML for *black box* applications. Modbus can be used in *full knowledge* applications with transparent gateway, but is not suitable because of the simple network structure.

Because of the termination of the application layer in *full knowledge* applications without transparent gateway, it is very much alike *black box* applications. OPC, MODBUS and M2MXML are therefore suitable as standard for the gateway-to-server communication. In theory, it is possible to use all the other standards, except from M2MXML, in the device network behind the gateway.

The following chapters describe possible solutions within the segments of AMR, Security and Utility Control based on the applications and solutions introduced above. It is some times desirable to collect data from more than one system behind a gateway; proposed solutions for such special type of applications are described in chapter 5.4.

5.1 - AMR

Figure 5.1.1 shows the standards applicable for Automatic Meter Reading. The standards at the top of the line are standards used for server-to-gateway communication, and those below are used for gateway-to-device communication.

Plain AMR applications can use DLMS/COSEM or LON as a complete server-to-device architecture. They can also use M-BUS or ZigBee in the gateway-to-device communication, combined with either one of the server-to-gateway standards shown in figure 5.1.1. Using OPC, M2MXML or MODBUS as server-to-gateway standards may be the obvious solution, but combining for example LON or DLMS/COSEM with M-BUS or ZigBee brings up some interesting alternatives. A network of M-BUS or ZigBee meters may act as a DLMS/COSEM or LON node, making it possible to integrate them with existing DLMS/COSEM or LON based AMR solutions.

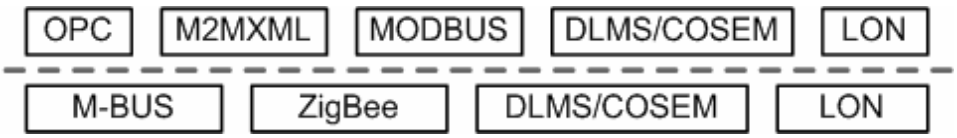

Figure 5.1.1 - Standards applicable for AMR

5.2 - Security

Figure 5.2.1 shows the standards applicable for Security applications. The standards at the top of the line are standards used for server-to-gateway communication, and those below are used for gateway-to-device communication.

LON or KNX might be used in complete server-to-gateway applications within security, depending on the level of security needed. Since ZigBee is the only standard with comprehensive security architecture it can be used in combination with OPC, M2MXML or MODBUS as long as the IP communication is secured beyond what those standards specifies.

OPC is a collection of standards, and one of them is OPC Alarm & Event. This standard can be used in the server-to-gateway communication in a security application making the gateway an OPC Alarm & Event server. This enables the M2M server in the M2M platform to browse and subscribe to alarm and events the system behind the gateway produces regardless of the underlying architecture, which can be ZigBee, LON, KNX, etc.

SIA and similar proprietary standards is the de facto within the security segment today. These standards can not be used as is in modern M2M solutions. However, SIA can be used to form new application layers in existing standards, as for example in a ZigBee security profile.

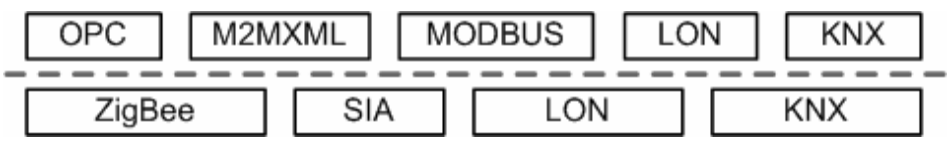

Figure 5.2.1 - Standards applicable for Security

5.3 - Utility Control

Figure 5.3.1 shows the standards applicable for Utility Control. The standards at the top of the line are standards used for server-to-gateway communication, and those below are used for gateway-to-device communication.

Utility Control is, in this thesis, automation systems extended to support M2M. Therefore, all the standards applicable for automation systems, within this thesis, are also applicable for the utility control segment, as long as they describe how to communicate using IP. The exception is ZigBee, which need another standard to transport the data over IP.

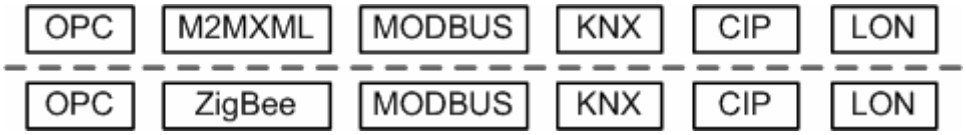

Figure 5.3.1 - Standards applicable for Utility Control

5.4 - Multiple services behind one gateway

The concept of multiple services behind on gateway is the ability to integrate different end-device networks in one M2M solution, using only one standard in the server-to-gateway communication. This will ease the development of the server since it only needs to support one standard. Each gateway knows how to talk to its end-devices and translates the communication to the selected server-to-gateway standard.

There are many M2M solutions using this concept today, but they often use proprietary solutions for either or both the server-to-gateway and gateway-to-device communication.

Figure 5.4.1 shows the applicable standards for multiple services behind one gateway. One of the standards on top of the line, which is applicable to carry any for of information, is selected. The gateways then support the selected server-to-gateway standard and the desirable gateway-to-device standards. Figure 5.4.1 does therefore include all gateway-to-device standards covered by this thesis.

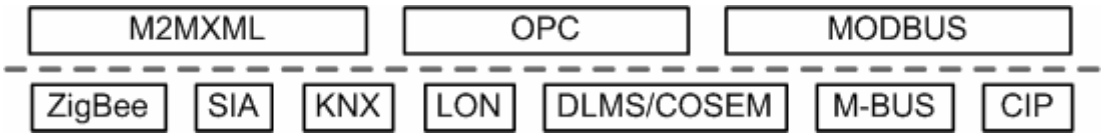

Figure 5.4.1 - Standards applicable for multiple services behind one gateway

6 - Discussion

This chapter discusses the applicable standards based upon the area of application and proposed solutions.

6.1 - AMR segment

AMR applications are all about gathering data from meters relatively fare away from the data collection system. This makes AMR suitable for M2M based solutions, and many companies therefore offer such solutions. Echelon, the company behind LonWorks, has developed their own LON based AMR platform called NES. There area also a number of other companies offering M2M based AMR solutions, such as Telenor Cinclus, Enmeret and Kamstrup. Many of today's AMR solutions use proprietary protocols, which prevent a customer from buying meters and collection systems from different providers. It also prevents the customer from changing meter supplier, or changing the collection system regardless of meters used.

There are mainly three standards directly suitable for M2M based AMR applications M-BUS, DLMS/COSEM and LonWorks NES. NES is a complete AMR platform, and differs from the other standards because of the line of products provided by Echelon. LonWorks and DLMS/COSEM are full server-to-device standards, and can therefore be used alone to form M2M based AMR application, as opposed to M-BUS which needs another standard to perform the server-to-gateway communication. M-BUS and DLMS/COSEM are most likely to be used by solution providers in development of their own AMR applications, and NES is more of a complete platform sold by Echelon. Even though, it is still possible to develop own AMR applications based upon LonWorks and the LonTalk protocol.

Among the applicable standards found for AMR, none of them is especially suitable for wireless applications. The solution for this might be to develop a new AMR profile for ZigBee. This profile can be based upon M-BUS or DLMS/COSEM. A ZigBee profile based upon DLMS/COSEM enables interesting solutions. This profile might enable DLMS/COSEM communication from server-to-device with a transparent application layer. Due to the structure of ZigBee there has to be an address translation in the gateway, though.

6.2 - Security segment

Within the security segment today's solutions tends to use the old SIA standards and its similar standards, or they use proprietary solutions. This may be a result of scepticism or the fact that many of the standards, which are found applicable for modern M2M solutions in this thesis, are lacking comprehensive security architectures. Either way, this segment needs more standardisation work in the future to provide secure and reliable standards that the security industry can trust. ZigBee might be one-step in the right direction for the home market, but wireless communication might not be desirable in more robust security applications. For an M2M-solution provider, such as Teleca, one possibility is to interconnect with existing applications using a black box approach with OPC, MODBUS or M2MXML as the server-to-gateway standard over a secure IP channel.

6.3 - Utility Control segment

The Utility Control segment is quite a large segment including all form of automation systems with M2M support. Such systems are often used as the classic M2M example with the ability to collect data from a remote site, and control the devices within the automation system. Due to this large coverage of systems, it is impossible to determine which standards that is more applicable than others, since it will depend on the automation system. All types of automation systems are therefore applicable for this segment, as mentioned in chapter 5.3. Even standards which do not describe how to transfer the application layer using IP, because this functionality can be added to the automation system by introducing a gateway and for example OPC, MODBUS or M2MXML.

6.4 - Multiple services behind one gateway

One M2M solution provider may support many M2M solutions for different costumers and different applications in one platform. For such type of platforms, a standard gateway supporting multiple services behind one gateway might be of interest. This raises some issues of discussion.

It is possible to use one standard in the server-to-gateway communication, regardless of the applications behind the gateway. This has advantages in form of only one standard to implement on the M2M server, and the M2M server developers only need to learn this standard. On the other hand, this solution depends on available gateways that support the selected standard and the standard used behind the gateway in the gateway-to-device communication. The platform also misses the features represented in the different standards, since it is locked to one.

The other solution is to support different standards in the server-to-gateway communication, depending on the application implemented in the platform. The server developers have to study and implement the different standards. In return, the platform gets a richer set of features to select from, due to the difference of the standards.

6.5 - Bandwidth consumption

Many M2M applications use communication bearers that charges based upon amount of transferred data. To these applications, the amount of transported data is of interest, since reducing the amount of bandwidth consumed will also reduce the total cost of the application. The tendency today is that these communication bearers is adapting to M2M and charges a fixed fee. However, this is still a topic of interest, since it will take time before all these communication bearers will offer a fixed fee.

Solutions with a transparent gateway and no filtering or routing have to produce more bandwidth than other solutions, since there is no logic that decides which data to forward or not. A router will reduce the amount of data since it only forwards the necessary data. Full featured networks, such as LON, CIP and KNX, needs the gateway to act as a router to reduce the amount of data passing through it. If not, all the data generated by devices in the local

device network will be sent to the M2M server using our amount charged communication bearer.

DLMS/COSEM differs from the other standards with a transparent application layer because of its server/client architecture. It is the end-devices that act as the server, meaning that the client (M2M server) initiate the communication and inquires data from the server (meter). DLMS/COSEM also specifies that the meters cannot communicate with each other, which prevents uninteresting data, from the M2M server's point of view.

By using a black-box view of the application, we can decide which server-to-gateway standard to use, regardless of the devices behind the gateway. This makes it possible to compare and decide. In this thesis three standards are found applicable for use with black-box applications; M2MXML, OPC and MODBUS. M2MXML is clearly the most unsuitable standard since it is based upon XML. Since OPC is a collection of different standards, the type of application will affect the amount of data produced. MODBUS has a simple and compact data format, since it is originally built for old serial line communication in the seventies. MODBUS is therefore most likely to consume least bandwidth.

6.6 - Software update

When deploying a large platform, automatic software update of the devices is often wanted. This saves time and money, as no service personnel needs to be on the site.

When using a gateway as a translator between two protocols the gateway has to take part in the software update. If the server-to-gateway standard does not support transfer of larger amount of data, FTP can be used in a proprietary solution. Then the standard for the end-device network is used to update the end-devices.

Standards such as CIP, LON and KNX support transportation of larger amount of data. This enables the ability to perform software updates passed the gateway. If one node in the device-network act as a second-hand gateway, i.e. gateway enabling for example a ZigBee network to act as a LON node. This gateway has to implement a software update procedure to enable software update of the second-hand devices.

If the standard selected does not support software update, and a proprietary solution is needed. The interoperability of the devices is tampered, and usage of off-shelf devices is no longer an option.

6.7 - Interconnection of networks

In a M2M platform, interconnection of networks might be required. Usage of networked standards, such as LON, KNX or CIP, enables interconnection of systems. But, they only support a relatively small infinite number of nodes. This will cause problems when interconnecting multiple networks. OPC on the other hand is built up only using the IP protocol, which enables free use of IP addresses without address translation. OPC is therefore a better alternative for an open standard when designing a large M2M platform that requires interconnection of networks.

7 - Conclusions

In general, it is possible to create standardised M2M solutions based upon existing standards within the segments of Security, AMR and Utility Control. Some standards can be used as is, while others need to be used in combination with another standard to fit in to the M2M platform. Utility Control and AMR has most suitable standards. The security segment needs more standardisation work to support full featured M2M based solutions.

There exist M2M based AMR solutions today, but many of them uses proprietary solutions. This prevents interchangeable applications that the market can benefit from. DLMS/COSEM and LonWorks can be used as is for AMR applications. M-BUS is also applicable, but it needs to be combined with OPC, MODBUS or M2MXML to fit in to the M2M platform. There are no standards that can be used as is to support standardised wireless AMR solutions, ZigBee might be the answer to this problem.

Within the security segment, there are almost no applicable standards, only a lot of proprietary solutions. This is probably because of scepticism and the lack of security implemented in existing applicable standards. A temporarily solution might be to use a black box approach and extend existing proprietary solutions to fit in to the M2M platform using OPC, MODBUS or M2MXML. SIA plays an important role within the security segment.

The Utility Control segment contains most applicable standard. This is due to the size of this segment. Utility Control consists of all automation systems, and all such standards are therefore applicable, even if they do not directly fit in to the M2M platform. Because full applicability can be reach in combination with OPC, MODBUS or M2MXML. Among the standards investigated are therefore KNX, CIP, LonWorks, ZigBee, MODBUS, OPC and M2MXML applicable for utility control.

Applications where the communication bearer is charged based upon amount of data transferred needs a router-enabled gateway to control the flow of the data. Full featured networks such as LON, CIP and KNX support this. DLMS/CODEM differs from the other standards with a transparent gateway since each meter acts as a server, and the collection system as a client. DLMS/CODEM meters cannot talk to each other, and the collection system has therefore full control of the bandwidth used. MODBUS consumes least bandwidth of the standards applicable to transfer any types of data.

OPC, MODBUS and M2MXML enable the support of transporting data from multiple services behind one gateway using one standard in the server-to-gateway communication.

When interconnecting networks, the address space might be a problem. OPC, which uses IP all the way, is therefore the most suitable standard for such applications.

Abbreviations

ADU	Application Data Unit
AMR	Automatic Meter Reading
APL	Application
APS	Application Support
AUC	Agder University College
CIP	Common Industrial Protocol
COSEM	COmpanion Specification for Energy Metering
CRC	Cyclic Redundancy Check
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CTDMA	Concurrent Time Domain Multiple Access
DLMS	Device Language Message Specification
DLMS-UA	DLMS User Association
DTMF	Dual-Tone Multi-Frequency
EDGE	Enhanced Data rates for [Global|GSM] Evolution
EHS	European Home Systems
EIB	European Installation Bus
EN	European Standards
FFD	Full Function Device
FTP	File Transfer Protocol
GPRS	General Packet Radio Service
GSM	Groupe Spéciale Mobile, Global System for Mobile communications
GW	Gateway
HDLC	High-Level Data Link Control
HES	Home Electronic System
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IR	Infra Red
ISDN	Integrated Services Digital Network

LAN	Local Area Network
M2M	machine-to-machine, machine-to-man, man-to-machine, machine-to-mobile and mobile-to-machine
MAC	Medium Access Control
M-BUS	Meter Bus
NES	Networked Energy Services
NWK	Network
ODVA	Open DeviceNet Vendor Association
OLE	Object Linking and Embedding
OPC	OLE for Process Control
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
PHY	Physical
PL	Power Line
PLC	Power Line Communication
RF	Radio Frequency (or wireless)
RFD	Reduced Function Device
SCPT	Standard Configuration Parameter Types
SIA	Securities Industry Association
SMS	Short Message Service
SNVT	Standard Network Variable Types
TCP	Transmission Control Protocol
TP	Twisted Pair
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
XML	Extensible Markup Language

References

[1] About: Echelon, the company
URL: http://www.echelon.com/company/default.htm

[2] F. Tiersch. LonWorks® Technology - An Introduction
Erfurt : Desotron Verlagsgesellschaft, 2000. ISBN: 3-932875-11-7

[3] Bonde, Mogens. Intelligente bygningsinstallationer med LonWorks.
Rødovre : El-Fagets Uddannelsesnævn, 1999. ISBN: 87-7832-044-5

[4] About: Echelon NES (Networked Energy Services)
URL: http://www.echelon.com/metering/

[5] About: Echelon Pyxos
URL: http://www.echelon.com/products/pyxos/

[6] About: Echelon i.Lon product series
URL: http://www.echelon.com/products/interfaces/default.htm

[7] ControlNet international and Open DeviceNet Vendor Association.
CIP Common Specification, Volume 1. Release 1.0 - June 5, 2001
URL: http://www.odva.org/

[8] ControlNet international and Open DeviceNet Vendor Association.
Ethernet/IP Adaptation of CIP Specification. Release 1.0 - June 5, 2001
URL: http://www.odva.org/

[9] Byron K. Appelt. M2MXML: An Open Standard for Machine-To-Machine Communications. SensorLogic, Inc.
URL: http://www.m2mxml.org/papers/m2mxml_white_paper.doc

[10] About: Sensor Logic, Inc.
URL: http://www.sensorlogic.com/aboutus.html

[11] SourceForge.net. What is SourceForge.net?
URL: http://sourceforge.net/docs/about/

[12] About: Securities Industry Association (SIA)
URL: http://www.sia.com/about_sia/

[13] Michael Jarosik, Dispelling the Myth. In M2M Magazine, September 2003
URL: http://www.m2mmag.com/print/back_article.asp?ARTICLE_ID=8

[14] Teleca, Teleca M2M Platform
URL: http://www.teleca.se/PSUser/mediacache/4561/4837/4856/M2M_Platform.pdf

[15] About: Open DeviceNet Vendor Association (ODVA)
URL: http://www.odva.org/ (click on the ‘About ODVA’ menu field)

[16] About: ControlNet International
URL: http://www.controlnet.org/01_abcn/index.htm

[17] A. Papenheim, C. Bories and Dr. Ziegel. The M-Bus: A Documentation.
Fachbereich Physik, Uinversität-GH Paderborn. Version 4.8, November 11, 1997
URL: http://www.m-bus.com/files/mbdoc48.exe

[18] DLMS User Association. Excerpts from: COSEM Architecture and Protocols,
Green Book - 5th edition. DLMS UA 1000-2 : 2005.
URL: http://www.dlms.com/documents/Excerpt_GB5.pdf

[19] DLMS User Association. Excerpts from: Conformance Test Process,
Yellow Book - 2th edition. DLMS UA 1001-1 : 2002.
URL: http://www.dlms.com/documents/Excerpt_YB2.pdf

[20] About: DLMS-UA, (DLMS - User Association)
URL: http://www.dlms.com/en/organisation/geninfos.htm

[21] Wikipedia. High-Level Data Link Control (HDLC)
URL: http://en.wikipedia.org/wiki/HDLC

[22] Konnex Association. KNX System Architecture. July 2004
URL: http://www.konnex.org/downloads/03%20-%20KNX%20Standard/
KNX%20Standard%20Public%20Documents/KNX%20System%20Architecture.pdf

[23] About: Konnex Association
URL: http://www.konnex.org/association/index.php

[24] Microsoft. COM: Component Object Model Technologies.
URL: http://www.microsoft.com/com/default.mspx

[25] About: SIA (Securities Industry Association)
URL: http://www.sia.com/about_sia/

[26] Modbus-IDA. Modbus Application Protocol Specification. V1.1a – June 4, 2004.
URL: http://www.modbus-ida.org/docs/Modbus_Application_Protocol_V1_1a.pdf

[27] ZigBee Alliance. ZigBee Specification. ZigBee document 053474r06, version 1.0, 2004
URL: http://www.zigbee.org/en/spec_download/download_request.asp

[28] About: ZigBee Alliance
URL: http://www.zigbee.org/en/about/

[29] OPC Foundation. About OPC: What is OPC? (web-page)
URL: http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC

[30] OPC Foundation. About OPC: What is the OPC Foundation? (web-page)
URL: http://www.opcfoundation.org/Default.aspx/01_about/01_history.asp?MID=AboutOPC

[31] OPC Foundation. OPC Overview. Version 1.0, October 27, 1998
URL: http://www.opcfoundation.org/DownloadFile.aspx?CM=3&RI=1&CN=KEY&CI=282&CU=15

[32] Harbor Research, The Impact of “Us Versus Them” Thinking.
In Harbor's free "Currents" newsletter, Issue 50 - June 17, 2005
URL: http://harborresearch.com/website/index2.php?option=com_yanc&listid=1&
action=view&send=2005-06-17%2011:09:24&Itemid=31

Appendix A - Feature Overview

Feature	CIP	MOD-BUS	LON	KNX	DLMS/ COSEM	M-BUS	SIA	M2M-XML	OPC	Zig Bee
Common application layer for different types of underlaying networks	x	x	x	x	x					
Advanced network with routing etc.	x		x	x					x	x
Designed to transport any types of data		x						x	x	
Designed to transport specifi types of data	x		x	x	x	x	x			x
Support transfere of larger amount of data such as software updates.	x		x	x						
Comprehensive security architecture										x
Total number of adressable nodes	60	247	23.000	65.000	65.000	250	0 **	~ *	~ *	65.000

*** Using TCP/IP enabling almost infinite number of addresses depending on the structure of the network**

**** Not structured as a network**